Research Proposal
Knowledge Graph Analysis and Repair

Robert David[1,2][0000−0002−3244−5341]

Institute for Data, Process and Knowledge Management,
Vienna University of Economics and Business, Austria

Abstract. The Semantic Web provides standards for knowledge graphs (KGs), which have become popular for solving data heterogeneity problems in enterprises, since they allow for flexible data modelling and integration via linking of graphs. This flexibility requires technologies to ensure data quality and consistent state, such as the Shapes Constraint Language SHACL. However, SHACL does not provide the means to explain why constraint violations are caused and how the KG can be repaired to conform to the constraints. Also, repairs for a KG can come with a high number of different alternative choices to pick from, where we need a way to determine preferences in practice. Finally, knowledge in the KG itself can be statistically exploited for repairs to determine fresh values and preferred choices and to identify incorrect data from a real-world perspective. For these challenges, we aim to develop a system that combines logic-based repairs and data-driven analysis for a repair approach that concludes KGs towards a quality fix point. The approach will not only be defined at the formal level, but we also will provide a prototypical implementation for practical experiments, thereby positioning it at the intersection of theoretical and applied research. Use cases shall be provided from companies, projects and open data to better understand how repairs can be applied effectively in practice. With this work we contribute to improving the quality of KGs by providing intelligent knowledge graph repair.

Keywords: Knowledge Graphs · Shapes Constraint Language · SHACL · Data Repair · Data Quality · Logic Programming · Data-Driven Anaylsis · Hybrid AI

1 Motivation

Knowledge graphs (KGs) [12] are often used in enterprises for consolidation of heterogeneous data sources. The flexible approach of modelling graphs using the Resource Description Framework (RDF) makes it possible to link together various data sources and thereby integrate them into an enterprise knowledge graph (EKG) [11]. However, problems like inconsistencies and incomplete data can be introduced. These problems generally represent quality issues that need appropriate mechanisms to detect and manage them in practice.

The Semantic Web traditionally uses open world reasoning based on the standards RDFS and OWL [3], which has very limited use regarding the detection of inconsistencies and cannot express how data can be changed to solve quality issues. The later introduced Shapes Constraint Language SHACL [13] can detect data inconsistencies based on constraints and returns a report about violations. However, these reports do not explain why a violation has happened and they do not provide sufficient information how to correct the data. As quality of graph data is of high importance in practice, having a system which can determine how to change graph data in such a way that is is repaired regarding defined constraints would be of high value.

In the following, we will present the state of the art for graph standards, ontology and database repairs and data-driven analysis. Then we continue with formulating research questions and contributions the thesis will provide. In the research methodology and approach section, we show how we plan to incrementally address the questions, followed by the evaluation plan. We then provide a report on the intermediary results up to now. Finally, we present conclusions and next steps.

2 State of the Art

The state of the art related to this thesis covers knowledge graph technologies, which are the foundation of our work. We also look into approaches to repair data for Ontologies and relational databases and the complexity of the repair problem. Finally, we need to understand the broad field of data-driven analysis methods, which can be used to determine repair choices.

2.1 Knowledge Graphs and the Semantic Web

The Semantic Web [3] is a set of technologies for knowledge representation for the world wide web. The basic structure of the Semantic Web is a knowledge graph, where nodes and edges are represented as resources on the web, thereby creating a world wide knowledge base. The Semantic Web includes languages to describe the semantics of (graph) data, RDF Schema RDFS and the Web Ontology Language OWL [3]. These languages allow for reasoning about the data in the form of logical conclusions. Because links to graphs can be created by anyone on the web, there was the need to design these logical conclusions with an open world assumption, applying monotonic reasoning and thereby avoiding problems with contradicting data. However, this means that constraints that be represented by these languages are limited to detecting inconsistencies in the Ontology regarding the graph data.

Enterprise Knowledge Graphs Originally created for open data publishing, the Semantic Web technologies were discovered to be sufficiently flexible and expressive to be used in enterprise contexts to solve data integration problems under the term Enterprise Knowledge Graphs (EKG) [11]. On the organisational side, EKGs differ from public knowledge graphs usually by being limited to the enterprise's data sources and the need of high quality to avoid false information which might become problematic from a business point of view. However, using flexible graph data models to manage complex semantic information in enterprise environments creates the challenge of quality control. Therefore, data quality of graph datasets needs to be defined in a principled manner in order to be able to detect conflicts and possibly repair them [6].

Graph Constraint Languages There are currently two constraint languages for knowledge graphs. The shapes constraint language SHACL [13] is currently the only W3C recommendation regarding constraint validation for the Semantic Web. The shape expressions language ShEx [14] is developed in a W3C community group and publicly available. SHACL and ShEx are both based around the idea of shapes that group together constraints to be evaluated on nodes in the knowledge graph. They differ in syntax, in some details regarding constraint validation and reporting, and they define different semantics for recursion. Semantics for shape validation are discussed in general in the scientific community regarding what makes sense in the scenario of data quality. There are different proposals regarding what the validation results should represent to be useful for data quality checks. Generally, the provided information should assist in fixing the data to achieve conformance with the constraints and thereby improve the quality of the graph data.

2.2 Repair Approaches

Ontology Repairs Related to our work are also reasoning tasks for explanations and repairs for Ontologies, often in the context of Description Logics, of which examples can be found in [7] and [8]. A recent work on Description Logic Ontologies is [4], which addresses optimal repairs in the scenario where the schema (TBox) is assumed to be correct, while the data (ABox) needs to be repaired. Optimal repairs in this context are repairs which preserve as much as possible from the logical consequences of the Ontology.

Relational Database Repairs Inconsistent data, as a consequence of violating integrity constraints, are well researched for relational databases. Two strategies for coping with inconsistent relational data are presented in [5]. First, consistent query answering (CQA) does not resolve the inconsistencies, but provides answers to queries based on a consistent subset of the data. Second, data cleaning repairs the inconsistencies while trying to preserve as much information as possible by doing minimal changes. This strategy can result in different choices how to change the data. One way of specifying and implementing repairs is to represent them as logic programs, e.g. using disjunctive Datalog with stable model semantics, which is described in [5]. These programs modify a database to achieve conformance with a set of integrity constraints.

2.3 Complexity and Scalability

When extending constraint validation with repairs, we have to consider the computational complexity of the repair problem, which means the amount of computationa1 resources (time or space) needed to solve it. For investigating the complexity of repairs, we look at complexity classes of the polynomial hierarchy [16]. A starting point is to look at the validation problem, which is querying if a data graph satisfies given constraints. The complexity to answer such a question depending on the size of the data graph and shapes graph is called combined complexity. The validation problem has already been shown to have an NP (solvable in polynomial time by a nondeterministic Turing machine) upper bound for combined complexity [10]. For the repair problem, we do not expect the combined complexity to be less, as validation can be reduced to it. This has to be considered when providing a solution that should be viable in practical scenarios.

2.4 Data-driven Analysis

Constraint languages are a formal way to detect inconsistencies in data sets. Implementing repairs will provide us with necessary changes to the data. However, coming up with fresh values that did not exist in the data before or picking from different choices for repairs cannot always be covered by formal constraints. A source of such knowledge is the data graph itself, which can be analysed to decide for fresh values, pick repair choices and to identify false data from a real-world perspective beyond the formal constraints. By combining such statistical analyses with crisp semantic descriptions from the data set, we can provide practically viable repair options.

This field of concluding new knowledge from sample data is very broad and there are many different approaches to do so, with the most prominent being machine learning methods, like Graph Neural Networks [15]. The field of Knowledge Graph Completion [9] focuses on predicting graph structures and values based on statistical analyses of existing graph data. It contains many different approaches, which range from probabilistic formal methods to modern machine learning techniques, like deep learning and large language models.

3 Problem Statement and Contributions

In the course of the thesis, we plan to address the following research questions and develop solutions for knowledge graph repair. This will be done theoretically on a formal level and also practically in the sense of applied research. On the theoretical side we provide novel formalisms to analyse graphs regarding repairs based on constraints. On the practical side, we provide prototypes that implement these formalisms. We then apply these prototypes to practical scenarios to gain further insights into how our approach can be used for repairs which are practically viable. With this work we aim to contribute to the advance of the academic discourse on knowledge graph quality.

We identify 3 main research questions regarding knowledge graph repairs. We also define associated sub research questions which contribute in answering the main research questions.

3.1 Explaining and repairing Constraint Violations

The first main research question is about how to define and implement repairs for graph data based on constraints. This is an open question in the research community that was not yet addressed. To limit the scope, our work focuses on SHACL as a constraint language.

The current constraint languages for knowledge graphs are designed to identify quality problems by specifying constraints grouped into shapes and then checking if the graph data satisfies them. The process of validation returns reports for constraint violations. However, reports do not provide an explanation for why a constraint is violated and how this can be solved. Providing an automatic solution for such violations is not trivial. Local solutions might interact with circular dependencies and thereby lead to an undecidable situation. Therefore explanations are non-deterministic in nature and have to be determined globally for a data graph. Currently it is open to how to compute such repairs for knowledge graphs.

To develop a solution, we first need a formal definition of repairs as explanations for non-validation of SHACL constraints. These explanations tell us how to change the data graph to achieve conformance with the constraints. We also need to discuss which kinds of explanations we would like to have from a practical point of view. For example, we can argue for minimality of changes, because we want to preserve as much of the original data graph as possible. Formalising these explanations is a preliminary step for providing an approach for computing and implementing repairs.

Second, we need an implementation using an appropriate technology and verify the correctness of the implementation. There are currently no standard methods to compute repairs for knowledge graph data. However, there are approaches for relational databases. We will investigate how to transfer these repair calculations to the knowledge graph data model and how to combine them with SHACL.

RQ-1: How can we compute repairs that correct graph data to conform to SHACL constraints and how can this be implemented?

- *RQ-1-a: What is our understanding of a SHACL repair in the sense of explaining non-validation for SHACL constraints in the context of database repairs and practical applicability.*
- *RQ-1-b: How can we implement SHACL repairs by using an existing technology which is appropriate for the repair problem.*

- *RQ-1-c: How well does the repair approach scale? What are the limitations? What does it mean for practical purposes?*

3.2 Repair Strategies

The second main research question continues from the first main research question and asks about ways to represent user preferences for the repairs on top of what SHACL repairs can express.

The motivation for this question comes from two open issues which are not addressed by the first question.

- First, users might decide for certain elements of the data graph that they should not be added or deleted by the repairs, which basically means that they are read-only. In a less restrictive situation the user might still allow these elements to be added or deleted, but it would not be a preferred situation to pick them if there are alternative choices to repair the data graph.
- Second, it can be the situation that there is a high number of possible and equally optimal repair choices, which makes it difficult for users to decide for one choice. For this situation, we let the user specify preferences for repairs, so that the repair implementation can reduce the number of optimal repairs automatically and thereby make it easier for the user to decide.

RQ-2: Based on our definition of a SHACL repair, we provide formalized repair strategies to users so that they can define optimal repair choices from a real world perspective as preferences for specific repair choices.

3.3 Data-driven repairs choices

The third main research question again picks up from the previous main research questions and brings in data-driven methods to SHACL repairs for determining repair preferences. The idea is to create a hybrid system of formal and data-driven approaches.

This question is motivated at first by the question how to come up with fresh values if a SHACL repair needs the addition of data. Such values usually cannot be determined by looking at the constraints and need to come from a different source. Generally, users can provide values manually when needed. However, proposing values automatically is desirable for better usability and to be less dependent on user knowledge. An approach to determine values is to take into account information from the (existing) data graph that can provide insights into how to choose values.

Also, we can use such an approach to determine which choices to preferably pick for repairs. Finally, we can check existing values and determine outliers which might be incorrect data. This use of statistical inference is applied to improve the quality of the data graph beyond the SHACL constraints.

To summarize the scenarios, we use data-driven methods for

- inferring fresh values in the case of additions to the data graph
- determining preferred choices in the case of multiple repair choices
- detecting outliers in the data graph to improve the quality

RQ-3: How can we formally integrate statistical inference into the repair strategies to conclude missing values, detect outliers and pick from multiple choices?

- *RQ-3a: How far does statistical inference improve the quality of a knowledge graph from a practical point of view?*

We note that research as part of this thesis will not be about new statistical inference methods, but rather how to integrate them with formal repairs. Statistical inference is a broad and well-researched field and we aim to make it better accessible and usable in practice for knowledge graph repairs.

Finally, we would like to point out other interesting research questions, which were identified, but will not be addressed in this work because of scope reasons.

- Are there alternative notions of explanations and repairs?
- How do repairs interact with ontological reasoning with RDFS and OWL?
- How can we address recursive dependencies of shape targets and repairs?
- Given the high complexity of the repair problem, how can we improve the scalability in practice?

These questions are future work outside of the scope limits of this thesis.

4 Research Methodology and Approach

The research methodology for this thesis addresses the combination of theoretical and applied research. We develop our work incrementally by starting with the formal basis on the theoretical part and then going towards implementation and experiments to apply our apprach to practical use cases.

4.1 Methodology

The initial question of this thesis is how we can identify and repair quality problems in enterprise knowledge graphs. We start with a gap analysis regarding this question and identify open issues that we have to address. We determine the main research questions based on a high level understanding of this problem. The research questions are stated to clearly define and narrow down the scope in this broad field of knowledge graph quality. The main research questions of this thesis are addressed incrementally, with each answer to a question providing the context for the next question. At the beginning we address *RQ-1* and provide a formal basis and a prototypical implementation. *RQ-2* and *RQ-3* have a focus on applied research in the context of specific use cases, where we can verify the applicability of our approach and incorporate the insights into our work.

- For *RQ-1*, we investigate how to calculate repairs for SHACL constraint violations, while considering practical aspects, like minimal changes done by repairs to preserve as much as possible of the existing data. We also discuss the computational complexity of repairs and choose an appropriate technology for implementation.
- For *RQ-2*, we consider the scenario where users want to provide preferences for different repair choices. *RQ-2* is answered in the context of use cases, where we specifically answer the question of what a user would accept as a high quality repair in such a scenario.
- For *RQ-3*, we look into data-driven methods for statistically inferring values in knowledge graphs as an extension of the outcomes of *RQ-2*. Answering *RQ-3* will allow us to integrate formal repairs with data-driven analysis methods and we will provide a prototypical implementation.

This concludes our work, which provides a system for knowledge graph repair built on a strong formal basis and developed in practical scenarios. To the best of our knowledge, our approach is novel in the way how we address data quality for knowledge graphs and our implementation is an innovation where no similar solutions currently exist.

5 Evaluation

The approach developed in this work is intended to be applied in practice. Therefore we will select several real-world use cases and evaluate how well our approach can be used to solve them. An important question is to understand what these use cases accept as a valid repair in practice in the sense that it should improve the quality of the data graph from a real-world perspective. We then evaluate if such repairs can be formally represented by our approach and if they can be implemented with an acceptable performance by using the developed prototype. Currently, we are looking into two use cases, where we explore what it means to repair the data in such a way that it satisfies the use case requirements. These use cases are:

- Automatic processing of contract data, which needs data consistency to satisfy legal requirements originating in the GDPR [17].
- Modifying an existing OWL ontology [3] to satisfy a given use case specific fragment, while preserving as much as possible of the original ontology in a syntactic and semantic sense.

6 Results

The research questions are intended to incrementally build up a knowledge graph repair system. We will publish intermediary results and provide implementations for proof of concept and practical experiments that complement the theoretical work and build up a unified analysis and repair system. Currently, we have published two intermediary results as conference papers.

- In *Reasoning about Explanations for Non-validation in SHACL* [1] we explain non-validation using the notion of a repair as a collection of additions and deletions whose application on a data graph results in a repaired graph that satisfies a set of SHACL constraints. This publications answers research question *RQ-1-a* and provides the foundation for the next steps, which build on the outcomes.
- In *Repairing SHACL Constraint Violations using Answer Set Programming* [2] we propose an algorithm to compute repairs by encoding the repair problem into an answer set program. We introduce several optimisations to the program that aims to maximise the benefit for practical scenarios. This publications answers research question *RQ-1-b*, builds on the previous outcomes and is the first step towards practical experiments.

With these two contributions we already gained insights into the topic of this thesis and shared them with the research community.

7 Conclusions and next Steps

Our work contributed to research by providing a formal foundation and prototypical implementation to compute repairs for knowledge graphs based on SHACL constraints. The next steps will cover user preferences and integration of data-driven methods.

- For user preferences, we intend to publish the results with one publication for each use case, where we share insights what kind of repairs users would accept as high quality repairs. Together with the publication, we will also publish the prototypical implementation for definable preferences.
- For integrating data-driven methods, we still need to identify an appropriate use case, which provides a data set where we can identify repair choices based on statistical analysis. Based on this scenario, we will research ways of integration into the repair system for a hybrid approach. Again, we intend to do a publication and a prototype.

With our work, we contribute to improving the quality of large data graphs, especially in an enterprise knowledge graph scenario, by providing intelligent knowledge graph repair.

Acknowledgments. This Ph.D. is done under the supervision of Axel Polleres (Ph.D. advisor, Vienna University of Economics and Business) and Shqiponja Ahmeta (Vienna University of Technology).

References

1. Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Simkus, M.: Reasoning about explanations for non-validation in SHACL. In: Bienvenu, M., Lakemeyer, G., Erdem, E. (eds.) Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021. pp. 12–21 (2021). https://doi.org/10.24963/kr.2021/2

2. Ahmetaj, S., David, R., Polleres, A., Šimkus, M.: Repairing shacl constraint violations using answer set programming. In: The Semantic Web – ISWC 2022: 21st International Semantic Web Conference, Virtual Event, October 23–27, 2022, Proceedings. p. 375–391. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-19433-7_22
3. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edn. (2011)
4. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Optimal abox repair w.r.t. static el tboxes: From quantified aboxes back to aboxes. In: The Semantic Web: 19th International Conference, ESWC 2022, Hersonissos, Crete, Greece, May 29 – June 2, 2022, Proceedings. p. 130–146. Springer-Verlag, Berlin, Heidelberg (2022). https://doi.org/10.1007/978-3-031-06981-9_8
5. Bertossi, L.: Database repairing and consistent query answering. Synthesis Lectures on Data Management **3**(5), 1–121 (2011)
6. Bonifati, A., Fletcher, G., Voigt, H., Yakovets, N., Jagadish, H.V.: Querying Graphs. Morgan & Claypool Publishers (2018)
7. Calvanese, D., Ortiz, M., Simkus, M., Stefanoni, G.: Reasoning about explanations for negative query answers in DL-Lite. J. Artif. Intell. Res. **48**, 635–669 (2013). https://doi.org/10.1613/jair.3870
8. Ceylan, İ.İ., Lukasiewicz, T., Malizia, E., Molinaro, C., Vaicenavicius, A.: Explanations for negative query answers under existential rules. In: Proc. of KR 2020. pp. 223–232 (2020). https://doi.org/10.24963/kr.2020/23
9. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge graph completion: A review. IEEE Access **8**, 192435–192456 (2020)
10. Corman, J., Reutter, J.L., Savkovic, O.: Semantics and validation of recursive SHACL. In: Proc. of ISWC'18. Springer (2018). https://doi.org/10.1007/978-3-030-00671-6_19
11. Galkin, M., Auer, S., Vidal, M.E., Scerri, S.: Enterprise knowledge graphs: A semantic approach for knowledge management in the next generation of enterprise information systems. In: ICEIS (2). pp. 88–98 (2017)
12. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., Ngomo, A.N., Polleres, A., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge, Morgan & Claypool Publishers (2021). https://doi.org/10.2200/S01125ED1V01Y202109DSK022
13. Knublauch, H., Kontokostas, D.: Shapes constraint language (SHACL). Tech. rep., W3C (Jul 2017), https://www.w3.org/TR/shacl/
14. Prud'hommeaux, E., Labra Gayo, J., Solbrig, H.: Shape expressions: An rdf validation and transformation language. ACM International Conference Proceeding Series **2014** (09 2014). https://doi.org/10.1145/2660517.2660523
15. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The Graph Neural Network Model. ieeetnn **20**(1), 61–80 (2009). https://doi.org/10.1109/TNN.2008.2005605
16. Stockmeyer, L.J.: The polynomial-time hierarchy. Theoretical Computer Science **3**(1), 1–22 (1976)
17. Wolford, B.: General Data Protection Regulation (GDPR). Available online: https://gdpr.eu/what-is-gdpr/ (2022), accessed on 20 July 2022